FEISTY calling tree / flow chart

* *Make.m* – select the simulation to run (global vs. certain locations; forcing file/timeframe)
  + Options
    - *Locs\_CESM\_4p4z\_spinup*() – 200 yr spinup using 1st year of 4p4z forcing at specific locations
    - *Spinup\_cesm*() – 200 yr spinup using 1st year of global FOSI forcing
    - *Spinup\_4p4z*() – 200 yr spinup using 1st year of global 4p4z forcing using z3 and z4 as distinct prey
    - *Spinup\_4p4z\_comb*() – 200 yr spinup using 1st year of global 4p4z forcing combining z3 and z4 into one prey group
    - *Locs\_CESM\_4p4z*() – Full time series of 4p4z forcing at specific locations
    - *FOSI\_cesm*() – Full time series of global FOSI forcing saving biomass and prodcution
    - *FOSI\_cesm\_catch*() – Full time series of global FOSI forcing saving fisheries yield (Matlab constraints on memory/storage prevented combing this with other run)
    - *CESM\_4p4z*() – Full time series of global 4p4z forcing using z3 and z4 as distinct prey
    - *CESM\_4p4z\_comb*() – Full time series of global 4p4z forcing combining z3 and z4 into one prey group
  + *FOSI\_cesm*()
    - *make\_parameters\_1meso*() – sets parameter values
      * output variable “param”
    - load the grid for the forcing file
    - add number of grid cells to parameter structure
    - set number of years
    - set number of days per year
    - set number of days per month
    - set the experiment name
    - [fname,simname] = *sub\_fname\_cesm\_fosi\_exper*(param,exper)
      * simname – specific filename from parameters
      * fname – directory plus filename indicating the forcin
    - create internal Matlab storage variables of size: grid cells x days
    - set the name of the spinup run that used the same forcing
    - load initial biomasses from the end of the spinup run
      * “sizetype*.*bio” (g/g/m2/d)
    - [Sml\_f,Sml\_p,Sml\_d,Med\_f,Med\_p,Med\_d,Lrg\_p,Lrg\_d,BENT] = *sub\_init\_fish*(ID,Sml\_f,Sml\_p,Sml\_d,Med\_f,Med\_p,Med\_d,Lrg\_p,Lrg\_d,BENT) – initializes the following state variables:
      * td – fraction of time in the pelagic
      * met – mass-specific metabolic rate (g/g/m2/d)
      * enc\_f – mass-specific encounter rate with forage fish (g/m2/d)
      * enc\_p – mass-specific encounter rate with large pelagic fish (g/m2/d)
      * enc\_d– mass-specific encounter rate with demersal fish (g/m2/d)
      * enc\_zm – mass-specific encounter rate with mesozooplankton (g/m2/d)
      * enc\_zl – mass-specific encounter rate with macrozooplankton (g/m2/d)
      * enc\_be – mass-specific encounter rate with benthos (g/g/m2/d)
      * con\_f – mass-specific consumption rate of forage fish (g/g/m2/d)
      * con\_p – mass-specific consumption rate of large pelagic fish (g/g/m2/d)
      * con\_d – mass-specific consumption rate of demersal fish (g/g/m2/d)
      * con\_zm – mass-specific consumption rate of mesozooplankton (g/g/m2/d)
      * con\_zl – mass-specific consumption rate of macrozooplankton (g/g/m2/d)
      * con\_be – mass-specific consumption rate of benthos (g/g/m2/d)
      * I – mass-specific total ingestion rate (g/g/m2/d)
      * die – predation mortality rates (g/m2/d)
      * pred – mass-specific predation mortality rates (g/g/m2/d)
      * nmort – natural mortality rates (1/d)
      * prod – productivity (g/m2/d)
      * nu – mass-specific energy available for growth or reproduction (g/g/m2/d)
      * gamma – mass-specific maturation rate to the next size class (g/g/m2/d)
      * rep – mass-specific reproduction rate (g/g/m2/d)
      * rec – recruitment into the size class (g/m2/d)
      * clev – consumption level as a fraction of maximum consumption (unitless)
      * caught – fishing yield (g/m2/d)
      * fmort – mass-specific fishing rate (g/g/m2/d)
    - create netcdf files
    - set initial month to zero
    - loop over years
      * load that year’s forcing – variable name “ESM”
      * loop over days – variable name “DY”
        + [Sml\_f, Sml\_p, Sml\_d, Med\_f, Med\_p, Med\_d, Lrg\_p, Lrg\_d, BENT, ENVR] = *sub\_futbio\_1meso\_mzpref*(DY, ESM, GRD, Sml\_f, Sml\_p, Sml\_d, Med\_f, Med\_p, Med\_d, Lrg\_p, Lrg\_d, BENT, param)
        + store daily values
      * calculate and store monthly means
    - close netcdf files
* *sub\_futbio\_1meso\_mzpref*(DY, ESM, GRD, Sml\_f, Sml\_p, Sml\_d, Med\_f, Med\_p, Med\_d, Lrg\_p, Lrg\_d, BENT, param)
  + get ESM forcing for that day
    - ENVR = *get\_ESM\_1meso*(ESM,GRD,param,DY) – extracts the forcing values for that day
    - bio = *sub\_neg*(bio) – checks for negative biomass values and sets to a small number of present
  + update benthic biomass from last time step (diff eq)
    - [BENT.mass,BENT.pred] = *sub\_update\_be*(BENT.mass, param, ENVR.det, [Md.con\_be,Ld.con\_be], [Md.bio,Ld.bio])
    - BENT.mass = *sub\_check*(BENT.mass)
  + set time spent in the pelagic based on pelagic-demersal coupling option – time spent is proportional to prey biomasses, which are different for large pelagics and demersals
    - Lp.td = *sub\_tdif\_pel*(ENVR. H, param, Mf.bio, Mp.bio, Md.bio)
    - Ld.td = *sub\_tdif\_dem*(ENVR. H, param, Mf.bio, Mp.bio, Md.bio, BENT.mass)
  + call metabolism function for each size and functional type
    - Sf.met = *sub\_met*(ENVR.Tp,ENVR.Tb,Sf.td,param.M\_s,param)
  + call encounter function for each prey type of each size and functional type
    - Sf.enc\_zm = *sub\_enc*(param, ENVR.Tp, ENVR.Tb, param.M\_s, ENVR.Zm, Sf.td, Sf.td, param.MZ)
  + call consumption function for each prey type of each size and functional type
    - Sf.con\_zm = *sub\_cons*(param, ENVR.Tp, ENVR.Tb, Sf.td, param.M\_s, Sf.enc\_zm)
  + call overconsumption function to prevent consumption rates greater than zooplankton loss rates and track how often it happens
    - [Sf.con\_zm,Sp.con\_zm,Sd.con\_zm,Mf.con\_zm,Mp.con\_zm,ENVR.fZm] = *sub\_offline\_zm*(Sf.con\_zm, Sp.con\_zm, Sd.con\_zm, Mf.con\_zm, Mp.con\_zm, Sf.bio, Sp.bio, Sd.bio, Mf.bio, Mp.bio, ENVR.dZm)
  + calculate total consumption by hand
  + call consumption level function
    - Sf.clev = *sub\_clev*(param,Sf.I,ENVR.Tp,ENVR.Tb,Sf.td,param.M\_s)
  + calculate predation rates by hand
  + calculate mass-specific predation rates by hand
  + call natural mortality rate function
    - Sf.nmort = *sub\_nmort*(param,ENVR.Tp,ENVR.Tb,Sf.td,param.M\_s)
  + call available energy and production function
    - [Sf.nu, Sf.prod] = *sub\_nu*(param,Sf.I,Sf.bio,Sf.met)
  + call maturation out of size class function
    - Sf.gamma = *sub\_gamma*(param.K\_l, param.Z\_s, Sf.nu, Sf.die, Sf.bio, Sf.nmort, 0, 0)
  + call reproduction function
    - [Mf.gamma,Mf.nu,Mf.rep] = *sub\_rep*(param.NX, Mf.gamma, Mf.nu, param.K\_a)
  + call recruitment function – larvae/small size class has its own function
    - Sd.rec = *sub\_rec\_larv*(Ld.rep,Ld.bio,param.rfrac)
    - Mf.rec = *sub\_rec*(Sf.gamma,Sf.bio)
  + call fish mass-balance function (diff eq)
    - Sf.bio = *sub\_update\_fi*(Sf.bio, Sf.rec, Sf.nu, Sf.rep, Sf.gamma, Sf.die, Sf.nmort)
  + call fishing rate function
    - [Mf.bio, Mf.caught, Mf.fmort] = *sub\_fishing\_rate*(Mf.bio, dfrate, param.MFsel)
  + call forward Euler check function – same check for negative biomasses as above
    - Sf.bio=*sub\_check*(Sf.bio)